PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Luc Van Brabant	Technology Center: 2100
Serial No.: 10/748,008 Confirm 6494	Group Art Unit: 2139
Filed: 12/30/2003	Examiner: Wang, Harris C.
For: ON-ACCESS AND ON-DEMAND DISTRIBUTED VIRUS SCANNING	Atty. Dkt. No.: 10830.0103.NP

APPEAL BRIEF TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Commissioner for Patents
PO Box 1450
Alexandria, Virginia 22313-1450

Sir:

This Appeal Brief is in support of Appellant's Notice of Appeal (Reinstatement) filed Aug. 14, 2008 in reply to the Official Action of May 14, 2008. Please apply the appeal brief fee of $510 previously paid on Feb. 26, 2008, and charge the additional fee of $30 (due to the FY 2009 fee increase) to EMC Corporation Deposit Account No. 05-0889. Please find submitted herewith a Fee Transmittal For FY 2009 form authorizing the charge of the additional fee of $30 and the charge of any other required fee for this Appeal Brief to EMC Corporation Deposit Account No. 05-0889.

I. REAL PARTY IN INTEREST

The real party in interest is EMC Corporation, by virtue of an assignment recorded at Reel 014859 Frame 0273.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

III. STATUS OF THE CLAIMS

Claims 1-28 have been presented for examination.

Claims 1-28 have been finally rejected, and are being appealed.

IV. STATUS OF AMENDMENTS

No amendment was filed after the final Official Action of Sept. 26, 2007 or after the subsequent Official Action of May 14, 2008 re-opening prosecution.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The appellant's invention relates generally to "on-access" virus scanning and "on-demand" virus scanning. "On-access" virus scanning occurs when a specified trigger occurs, such as when a user accesses a file marked "unchecked." (Appellant's specification, page 4, lines 7-8.) "On-demand" virus scanning is typically scheduled when a new virus is discovered, when new unchecked files are migrated to a file server, or prior to archiving or backing-up unchecked files. (Appellant's specification, page 4, lines 8-10.)

The appellant's invention of independent claim 1 provides a method of operating a plurality of virus checkers (32, 33, and 34 in FIG. 1 and FIG. 3) for on-demand anti-virus scanning concurrent with on-access anti-virus scanning. (Appellant's specification, page 4, line 21-23; page 9 lines 3-7). The method of claim 1 includes combining on-demand anti-virus scan requests and on-access anti-virus scan requests in a virus scan request queue (63 in FIG. 3), and distributing the on-demand anti-virus scan requests and the on-access anti-virus scan requests from the virus scan request queue to the virus checkers. (Appellant's specification, page 4, line 23 to page 5, line 2; page 13 lines 13-15; page 13 lines 1-7). For example, the on-demand anti-virus scan requests and the on-access anti-virus scan requests are distributed from the virus scan request queue to the virus checkers by AV threads in a pool (64 in FIG. 3), and each AV thread is programmed as shown in steps 71 to 83 of FIGS. 4 and 5 to send the request from the head of the virus scan request queue (step 76 in FIG. 4) to a particular one of the virus checkers (step 83 in FIG. 5) assigned by a distribution list (62 in FIG. 3), as described in appellant's specification on page 13 lines 1-4 and page 13 line 23 to page 14 line 21.

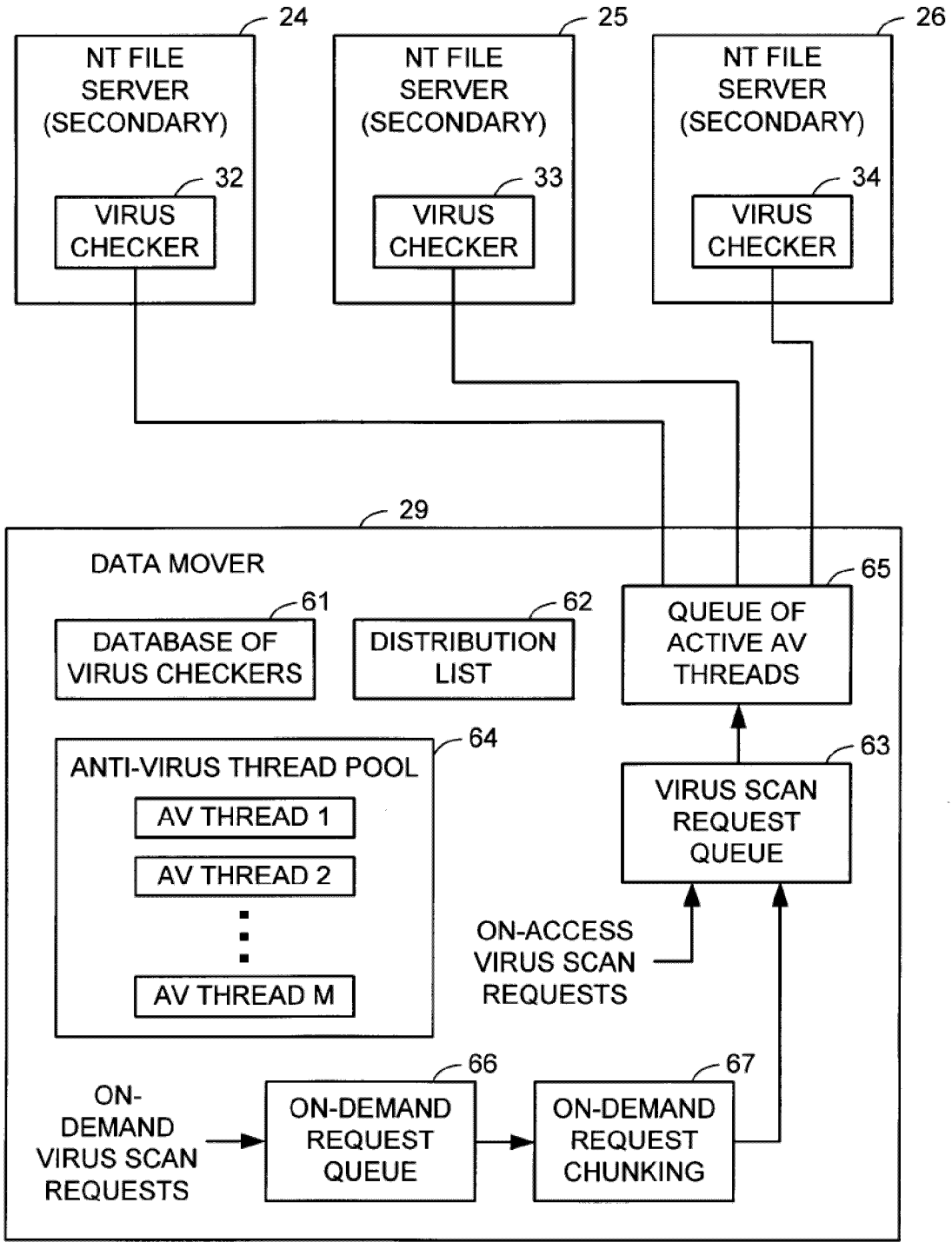
24
NT FILE SERVER (SECONDARY)
32
VIRUS CHECKER
25
NT FILE SERVER (SECONDARY)
33
VIRUS CHECKER
26
NT FILE SERVER (SECONDARY)
34
VIRUS CHECKER
29
DATA MOVER
61
DATABASE OF VIRUS CHECKERS
62
DISTRIBUTION LIST
65
QUEUE OF ACTIVE AV THREADS
64
ANTI-VIRUS THREAD POOL
AV THREAD 1
AV THREAD 2
AV THREAD M
63
VIRUS SCAN REQUEST QUEUE
ON-ACCESS VIRUS SCAN REQUESTS
66
ON-DEMAND REQUEST QUEUE
67
ON-DEMAND REQUEST CHUNKING
ON-DEMAND VIRUS SCAN REQUESTS

FIG. 3

The appellant's invention of independent claim 8 provides a method of operating a plurality of virus checkers (32, 33, and 34 in FIG. 1 and FIG. 3). (Appellant's specification, page 5, lines 3-4; page 9 lines 3-7.) The method includes distributing on-demand anti-virus scan requests and on-access anti-virus scan requests to the virus checkers so that the virus checkers perform on-demand anti-virus scanning concurrent with on-access anti-virus scanning. (Appellant's specification, page 5, lines 4-7.) The method further includes grouping the on-demand anti-virus scan requests into chunks of multiple ones of the on-demand anti-virus scan requests (67 in FIG. 3; steps 108-109 in FIG. 7), and for each chunk, distributing the multiple ones of the on-demand anti-virus scan requests over the virus checkers. (Appellant's specification, page 5, lines 7-10; page 13 lines 13-17 and 20-22; page 16, lines 18-22.)

The appellant's invention of independent claim 12 provides a method of operating a plurality of virus checkers (32, 33, and 34 in FIG. 1 and FIG. 3) for on-demand anti-virus scanning concurrent with on-access anti-virus scanning. (Appellant's specification, page 5, lines 11-13; page 9 lines 3-7.) The method includes combining on-demand anti-virus scan requests and on-access anti-virus scan requests in a virus scan request queue (63 in FIG. 3), and a pool of threads (64 in FIG. 3) distributing the on-demand anti-virus scan requests and the on-access anti-virus scan requests from the virus scan request queue to the virus checkers. (Appellant's specification, page 5, lines 13-16.) Each anti-virus scan request on the virus scan request queue is serviced by a respective one of the threads in the pool of threads. (Appellant's specification, page 5, lines 17-18.) The method further includes grouping the on-demand anti-virus scan requests into chunks of multiple ones of the on-demand anti-virus scan requests (67 in FIG. 3;

steps 108-109 in FIG. 7), and for each chunk, checking whether the number of anti-virus scan requests on the virus checking queue is less than a threshold (TH1 in step 124 of FIG 8), and upon finding that the number of anti-virus scan requests on the virus checking queue is less than the threshold, placing the chunk on the virus scan request queue (step 126 in FIG. 8). (Appellant's specification, page 5, lines 18-23; page 13 lines 13-17; page 17 line 21 to page 18 line 3.)

The appellant's invention of independent claim 16 provides a virus checking system including a plurality of virus checkers (32, 33, and 34 in FIG. 1 and FIG. 3) for on-demand anti-virus scanning concurrent with on-access anti-virus scanning, a virus scan request queue (63 in FIG. 3), and at least one processor (29 in FIG. 3) coupled to the virus checkers and the virus scan request queue for sending virus scan requests from the virus scan request queue to the virus checkers. (Appellant's specification, page 6, lines 2-6; page 9 lines 3-7; page 8 line 22 to page 9 line 1.) The at least one processor is programmed for placing on-demand anti-virus scan requests and on-access anti-virus scan requests onto the virus scan request queue, and for distributing the on-demand anti-virus scan requests and the on-access virus scan requests from the virus scan request queue to the virus checkers. (Appellant's specification, page 6, lines 6-10.)

The appellant's invention of independent claim 24 provides a virus checking system including a plurality of virus checkers (32, 33, and 34 in FIG. 1 and FIG. 3) for on-demand anti-virus scanning concurrent with on-access anti-virus scanning, and a file server (27 in FIG. 1) coupled to the virus checkers for sending virus checking requests from the file server to the virus checkers. (Appellant's specification, page 6, lines 11-15; page 8 lines 8-11 and 17-18; page 9

lines 3-7.) The file server includes a virus scan request queue (63 in FIG. 3). (Appellant's specification, page 6, line 15; page 12 lines 13-15; page 13, lines 2-7.) The file server is programmed for placing on-demand anti-virus scan requests and on-access anti-virus scan requests onto the virus scan request queue, and for executing multiple threads (64 in FIG. 3) for distributing the on-demand anti-virus scan requests and the on-access anti-virus scan requests from the virus scan request queue to the virus checkers (steps 71 to 83 of FIGS. 4 and 5). (Appellant's specification, page 6, lines 15-19; page 13 lines 2-17; page 13 line 23 to page 14 line 21.) Each anti-virus scan request on the virus scan request queue is serviced by a respective one of the threads in the pool of threads (steps 71 to 83 of FIGS. 4 and 5). (Appellant's specification, page 6, lines 19-21; page 13 line 23 to page 14 line 21.) The file server is further programmed for grouping the on-demand anti-virus scan requests into chunks (67 in FIG. 3; steps 108-109 in FIG. 7) of multiple ones of the on-demand anti-virus scan requests, and for consecutively placing the chunks onto the virus scan request queue. (Appellant's specification, page 6, lines 21-23; page 13 lines 13-17 and 20-22; page 16, lines 18-22.)

None of appellant's claims contain any "means plus function" or "step plus function" as permitted by 35 U.S.C. 112, sixth paragraph.

Appellant's dependent claims 5 and 21 further define giving the on-access anti-virus scan requests priority over the on-demand anti-virus scan request by inhibiting the placement of on-demand anti-virus scan requests onto the virus scan request queue (63 in FIG. 3) when the number of anti-virus scan requests on the virus scan request queue reaches a threshold (TH1 in step 124 of FIG 8), and not inhibiting the placement of on-access anti-virus scan requests onto

the virus scan request queue when the number of anti-virus scan requests on the virus scan request queue reaches the threshold. (Steps 124 and 126 in FIG. 8; appellant's specification, page 12, 13-20; page 13 lines 8-17; page 17 line 21 to page 18 line 3.) In a similar fashion, appellant's dependent claim 27 further defines checking for each chunk whether the number of anti-virus scan requests on the virus checking queue is less than a threshold (TH1 in step 124 of FIG 8), and upon finding that the number of anti-virus scan requests on the virus checking queue is less than the threshold, placing said each chunk on the virus scan request queue. (Steps 124 and 126 in FIG. 8; appellant's specification, page 12, 13-20; page 13 lines 8-17; page 17 line 21 to page 18 line 3.)

Appellant's dependent claims 6 and 22 further define grouping the on-demand anti-virus scan requests into chunks (67 in FIG. 3), each of the chunks including multiple ones of the on-demand anti-virus scan requests (steps 108-109 in FIG. 7), and placing the chunks onto the virus scan request queue (step 110 in FIG. 7; step 126 in FIG. 8). (Appellant's specification, page 12 lines 20-23; page 13 lines 13-17 and 20-22; page 16, lines 18-22; page 17, lines 12-14.)

Appellant's dependent claims 7, 11, 15, 23, and 28 further define inhibiting the placement of at least one of the chunks onto the virus scan request queue until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks (step 110 in FIG. 7; steps 121, 122, and 123 in FIG. 8). (Appellant's specification, page 12 lines 20-23; page 16, lines 18-22; page 17, line 12 to page 18, line 3.)

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether claims 1-3, 6-11, 16, 18-19, and 22-23 are unpatentable under 35 U.S.C. 102(e) as being anticipated by Muttik U.S. Patent Application Publication 2003/0046611.

2. Whether claims 4-5, 12-15, 17, 20-21, and 24-28 are unpatentable under 35 U.S.C. 103(a) as being unpatentable over Muttik U.S. Patent Application Publication 2003/0046611 over Edwards U.S. Patent 7,188,367.

VII. ARGUMENT

1. Claims 1-3, 6-11, 16, 18-19, and 22-23 are patentable under 35 U.S.C. 102(e) over Muttik U.S. Patent Application Publication 2003/0046611.

"For a prior art reference to anticipate in terms of 35 U.S.C. § 102, every element of the claimed invention must be identically shown in a single reference." Diversitech Corp. v. Century Steps, Inc., 7 U.S.P.Q.2d 1315, 1317 (Fed. Cir. 1988), quoted in In re Bond, 910 F.2d 831,15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990) (vacating and remanding Board holding of anticipation; the elements must be arranged in the reference as in the claim under review, although this is not an ipsis verbis test).

Muttik discloses a queue of files to be scanned, and a virus checker that sequentially scans the files in the queue. Muttik paragraph [0031] discloses that a server 4 performs regular on-demand anti-virus scans during quiet times. The scan is performed in a circular manner, such that when all of the files to be scanned have been scanned it starts again from the first file. The data defining the properties to be scanned for can be updated during the scan. (Muttik, Abstract.) If new files are added to the system mid-scan these are placed at a high position in the queue so that they are scanned soon. (Muttik, paragraph [0016].)

Claims 1, 3, 16, and 19

Appellant respectfully submits that Muttik does not disclose "combining on-demand anti-virus scan requests and on-access anti-virus scan requests in a virus scan request queue." In particular, Muttik fails to disclose that the adding of the new files to the system mid-scan and the placing of these new files at a high position in the queue so that they are scanned soon (Muttik page 2 paragraphs [0016] and [0037]) are on-access virus scan requests. Appellant further submits that Muttik does not disclose distributing virus scan requests from the virus scan request queue to a plurality of virus checkers.

With respect to claims 1 and 16, page 3 of the Official Action of May 14, 2008 cites Muttik paragraph [0037] for disclosing "If new files are added to the system during the scan, these are placed in a high position in the queue so that they are scanned soon." Page 4 of the Official Action of May 14, 2008 "interprets new files being added to the system as a scan request produced in response to user access to files, ..." Appellant respectfully disagrees with this interpretation. Muttik fails to disclose that such scans upon new files added to the system during a scan are produced in response to user access to the new files. Muttik paragraph [0037], for example, says:

> [0037] If new files are added to the system during the scan, these are placed in a high position in the queue, so that they are scanned soon. This is done by allocating a number such as N+1 or N+2 to the file, thereby ensuring that they are requested almost immediately. Any new files added to the system are a potential source of virus infection and as such an early scan is highly desirable.

In addition, both Muttik and the appellant's disclosure refer to scan requests for new files being added or migrated to a file system as being on-demand requests. Muttik paragraph 4, for example, says:

> [0004] It is known to provide anti-virus computer programs and E-mail and data filtering programs. Anti-virus programs may operate in an on-access mode or an on-demand mode. The on-access mode initiates a scan of a file when an access request to that file is made. The on-demand mode initiates a scan of all files on a specified volume or volumes either on a user request or on a scheduled request.

Therefore, the placement of a new file being added to the system at a high position on the virus scan request is seen as a scheduled request, for example as an incoming E-mail message flows into the system, as described in Muttik paragraph [0006].

Muttik's description of his preferred embodiments is also directed to on-demand virus scanning, rather than on-access virus scanning. Muttik paragraph [0031] begins: "In operation the network storage device 18 is subject to regular on-demand scans to identify computer viruses, Trojans, Worms and/or files with banned content." (Emphasis added.) Muttik paragraph [0032] says: "FIG. 2 shows a flow diagram illustrating the steps performed during a scan of computer files to detect files having predefined properties indicating specific content such as the presence of a computer virus. The scan is initialized either by a user request or it is triggered automatically by an event such as the computer being turned on. ..." Thus, the description of the initialization of a scan in Muttik's system of FIG. 2 tracks Muttik's definition of "on-demand" virus scanning in Muttik's paragraph [0004] . Muttik's description of the

initialization of a scan in his system of FIG. 4 is similar. Muttik's paragraph [0040] says: "The flow diagram shows how following initialization f a scan, either by user request or by a present condition being fulfilled, N is set to one and file N(1) is requested. …"

Muttik's definition and examples of "on-demand" virus scanning are consistent with the appellant's definition and examples. Paragraph 4 of the appellants' specification says:

> [0007] For virus checking of files in a file server, it is desirable to perform "on-access" virus scanning on a priority basis, and "on-demand" virus scanning in background. "On-access" virus scanning occurs when a specified trigger occurs, such a when a user accesses a file marked "unchecked". "On-demand" virus scanning is typically scheduled when a new virus is discovered, when new unchecked files are migrated into a file server, or prior to archiving or backing-up unchecked files.

Thus, according to the appellant's definition and examples, the scheduling of virus scans "when new unchecked files are migrated into a file server" are "on-demand" virus scan requests. In addition, the scanning in FIG. 2 and FIG. 4 of Muttik is done during non-working periods or quiet times, or at low or zero priority during normal working time. (Muttik paragraphs [0014], [0032], and [0038].) Thus, the scanning in Muttik FIG. 2 and FIG. 4 is done in background, in comparison to user access which occurs at high priority during normal working time, consistent with the appellant's definition of "on-demand" virus scan requests.

Nor would it be inherent or necessary for the new files disclosed in Muttik paragraph [0016] to be added to the system in response to user access to the new files. New files are often

added to a system well before they are accessed by a user. For example, it is common knowledge that e-mail and e-mail attachments (e.g., the "messages flowing in" as suggested in Muttik paragraphs [0004] and [0006]) are downloaded to a user's e-mail server or to a user's computer well before the e-mail and e-mail attachments are accessed by a user opening the e-mail and e-mail attachments. In addition, Muttik does not suggest that a scan request would be placed on his queue for a new file being added to his system in response to a user access request. In accordance with Muttik's definition and examples of on-access and on-demand anti-virus scans, Muttik suggests that if a user would like to access a file that has not been scanned, then the file should be scanned immediately, prior to the user accessing it.

Appellant's independent claims 1 and 16 further recites "a plurality of virus checkers for on-demand anti-virus scanning concurrent with on-access anti-virus scanning;" and "distributing the on-demand anti-virus scan requests and the on-access virus scan requests from the virus scan request queue to the virus checkers." Although Muttik's server 4 would have at least one processor (such as the central processing unit 202 of the general purpose computer 200 of FIG. 5 as mentioned in Muttik paragraph [0043]), it is not inherent that the at least one processor of the server 4 would distribute the on-demand ant-virus scan requests from the virus scan request queue to a plurality of virus checkers. Instead, Muttik FIGS. 2, 3, and 4 show that the virus scans are performed in sequence in a circular manner, such that when all of the files to be scanned have been scanned it starts again from the first file (see also Muttik, Abstract), so that there is no need for distributing the on-demand anti-virus scan requests from the virus scan

request queue to a plurality of virus checkers. The requested on-demand anti-virus scan requests could be performed by a single virus checker such as a virus checker program executed by the central processor of the server 4.

Claims 2 and 18

Appellant' dependent claims 2 and 18 are distinguished from Muttik for the reasons give above with respect to their respective base claims 1 and 16. In addition, claims 2 and 18 explicitly define that the on-access anti-virus scan requests are produced in response to user access to files. As discussed above with respect to claim 1, Muttik fails to disclose that the adding of the new files to the system mid-scan or the placing of these new files at a high position in the queue is done in response to user access to the new files.

Claims 6 and 22

Appellant's dependent claims 6 and 22 are distinguished from Muttik for the reasons given above with respect to their respective base claims 1 and 16. Appellant's dependent claims 6 and 22 further distinguishes Muttik by reciting "grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and placing the chunks onto the virus scan request queue.

Pages 4 to 5 of the Official Action cite Muttik FIG. 3 and paragraphs [0031] and [0037] for grouping the on-demand virus scans into chunks where the new file added to the queue in N+1 or N+2 splits the on-demand virus scans into chunks. However, in Muttik, the files before

N+1 or N+2 and the files after N+1 or N+2 are not distinguished or treated as groups. Instead, the files in the queue are scanned in sequence in a circular manner, such that when all of the files to be scanned have been scanned it starts again from the first file. (Muttik, FIG. 2 and Abstract.) Appellant's dependent claim 6 recites an additional operation of "placing the chunks onto the virus scan request queue." This additional operation is not done in Muttik. It is the new file added to the queue in the N+1 or N+2 position that is placed onto the virus scan request queue. The old files are already on the queue before the new file is added to the queue, so there is no operation of "placing the chunks onto the virus scan request queue."

Claims 7 and 23

Appellant's dependent claims 7 and 23 are distinguished from Muttik for the same reasons given above for their respective base claims 6 and 22. In addition, claims 7 and 23 further recite "inhibiting the placement of at least one of the chunks onto the virus scan request queue until completion of anti-virus scanning of the anti-virus scan requests in a prior one of the chunks." With respect to claims 7 and 23, page 5 of the Official Action cites Muttik paragraph [0037] for "inhibiting the on-demand scan's placement in the queue by first scanning files as a result of an access to a file." However, Muttik does not teach inhibiting the placement of at least one of the chunks onto the virus scan request queue until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks. Placing a new request at a high position in the queue before an earlier request already on the queue is different from inhibiting the placement of the earlier request onto the queue. (Moreover, as discussed above with respect to claims 1 and

16, appellant's view is that Muttik paragraph [0037] does not disclose scanning a file as a result of an access to the file.)

Claims 8 and 10

With respect to appellant's independent claim 8, as discussed above with respect to claims 1 and 16, Muttik does not disclose distributing on-demand anti-virus scan requests and on-access anti-virus scan requests to a plurality of virus checkers so that the virus checkers perform on-demand anti-virus scanning concurrent with on-access anti-virus scanning. Instead, Muttik discloses a queue of files to be scanned on-demand, and a virus checker that sequentially scans the files in the queue. (See, for example, Muttik paragraph [0031], in which the server 4 performs regular on-demand scans upon the network storage device 18 during quite times.) Muttik also does not disclose grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and for each chunk, distributing the multiple ones of the on-demand anti-virus scan requests over the virus checkers. Instead, in Muttik, the files in the queue are scanned in sequence in a circular manner, such that when all of the files to be scanned have been scanned it starts again from the first file. (Muttik, FIG. 2 and Abstract.)

Claim 9

Appellant's dependent claim 9 is distinguished from Muttik for the reasons given above with respect to its base claim 8. Claim 9 further defines that the on-access anti-virus scan requests are produced in response to user access to files. As discussed above with respect to

claims 1 and 2, Muttik describes on-demand virus scanning with reference to Muttik's FIGS. 1-5. Muttik mentions on-access anti-virus scanning in the Description of the Prior Art in paragraph [0004], but Muttik does not disclose performing on-demand anti-virus scanning concurrent with on-access anti-virus scanning. In particular, as discussed above, Muttik fails to disclose that the adding of the new files to the system mid-scan and the placing of these new files at a high position in the queue so that they are scanned soon (Muttik page 2 paragraphs [0016] and [0037]) are on-access virus scan requests. Muttik fails to disclose that the adding of these new files to the system mid-scan or the placing of these new files at a high position in the queue is done in response to user access to the files.

Claim 11

Appellant's dependent claim 11 is distinguished from Muttik for the reasons given above with respect to its base claim 8. Claim 11 further recites "inhibiting the distribution of the multiple ones of the on-demand anti-virus scan requests from at least one of the chunks to the virus checkers until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks." Therefore claim 11 is further distinguished from Muttik for reasons similar to the reasons discussed above with reference to appellant's claims 7. Muttik does not distribute on-demand anti-virus scans from chunks to a plurality of virus checkers. Nor does Muttik further disclose inhibiting the distribution of the multiple ones of the on-demand anti virus scan request from at least one of the chunks to the virus checkers until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks. Instead, Muttik discloses a

queue of files to be scanned on-demand, and a virus checker that sequentially scans the files in the queue in a circular manner, such that when all of the files to be scanned have been scanned it starts again from the first file. (See Muttik, FIG. 2, Abstract, and paragraph [0031].)

2. Claims 4-5, 12-15, 17, 20-21, and 24-28 are patentable under 35 U.S.C. 103(a) over Muttik U.S. Patent Application Publication 2003/0046611 in view of Edwards U.S. Patent 7,188,367.

The policy of the Patent and Trademark Office has been to follow in each and every case the standard of patentability enunciated by the Supreme Court in Graham v. John Deere Co., 148 U.S.P.Q. 459 (1966). See M.P.E.P. § 2141; KSR International Co. v. Teleflex Inc., 550 U.S. __, 82 USPQ2d 1385 (2007). As stated by the Supreme Court:

> Under § 103, the scope and content of the prior art are to be determined; differences between the prior art and the claims at issue are to be ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background, the obviousness or nonobviousness of the subject matter is determined. Such secondary considerations as commercial success, long felt but unsolved needs, failure of others, etc., might be utilized to give light to the circumstances surrounding the origin of the subject matter sought to be patented. As indicia of obviousness or nonobviousness, these inquiries may have relevancy.

148 U.S.P.Q. at 467.

Scope and Content of the Prior Art

Muttik has been summarized above with respect to the rejections under 35 U.S.C. 102.

Edwards discloses a virus scanner in which a pool of pre-processor threads and a queue are interposed between the event filter and the pool of scanner threads. The pre-processor threads perform operations that can be completed quickly to determine whether an object of a scan request needs to be scanned. The pre-processor threads gather characteristics about the scan requests and place them in the queue in a priority order based on those characteristics. The scanner threads select a scan request from the queue based on the priority order. Alternatively, the scan request is selected based on the scan request's characteristics as compared to the characteristics of the scan requests whose objects are currently being scanned by other scanner threads in the pool. (Edwards, Abstract.) Virus scanners may be invoked on-demand by a computer user to scan a selected file. More typically, virus scanners install themselves as part of an operating system, and then scan files, according to user preferences, as the files are created and accessed. This type of virus scanner is referred to as an on-access virus scanner. (Edwards, col. 1, lines 42-47.)

Differences between the prior art and the claims at issue

Differences between Muttik and the limitations recited in appellant's independent claims 1, 8, and 16 have been discussed above with respect to the rejections under 35 U.S.C. 102.

With respect to the limitations recited in appellant's claims 1, 8, and 16, appellant respectfully submits that the system shown in FIG. 1 of Edwards and labeled "Prior Art" is said to be an on-access virus scanner. Edwards, col. 2, lines 12-24, say:

> A prior art on-access virus scanner 100 is organized into two parts, as illustrated in FIG. 1. One part is the event filter 110, which is the software that intercepts the events of interest to the virus scanner. Events of interest include a file being opened or an e-mail arriving in a mailbox. Another part is a scanner thread 120, which is the software that receives scan requests from the filter. The scanner thread determines whether the object of the intercepted event (i.e. the file, e-mail, or e-mail attachment) needs scanning and, if so, scans the object. Multiple scanner threads are typically provided in pools 130 that are capable of executing concurrently so that multiple objects may be scanned simultaneously.

As further disclosed in Edwards column 2, lines 25-37, Edwards is directed to a problem that arises in the on-access virus scanner:

> Unfortunately, virus developers have recently begun to manufacture "malicious" files which take "a long time" to scan, including archives and documents containing embedded objects. The malicious files are designed to overwhelm on-access virus scanners by tying up all of the available scanner threads in the pool, thereby causing all other events intercepted by the filter to be queued until a scanner thread becomes free. This causes the virus scanner to

> "crash" by blocking further processing of data and leaves a system undefended against subsequent attacks. If e-mail or file processing is routed through a virus scanner and the scanner has crashed, then a "denial of service" for e-mail or file activity occurs until the scanner is restarted.

Edward's embodiment disclosed in FIGS. 2-6 also appears to be directed to an on-access virus scanner because the disclosed embodiment includes an event filter (Edwards, FIG. 2, box 110). Edwards col. 4, lines 50-53 say: "A scan request is generated whenever an event occurs that causes the host system to access an object, such as opening a file, reading an e-mail, or opening an e-mail attachment." Edwards col. 4 lines 64-67 say: "As illustrated, a pre-processor pool 230 of four pre-processor threads 210 and a priority queue 220 are interposed between the event filter 110 and the scanner thread pool 130 of three scanner threads 120." Edwards col. 10, lines 3-7 say: "For example, while the foregoing description focused on on-access virus scanners, it will be recognized that the above techniques and analyses can be applied to scanning data in other contexts such as on-demand virus scanners having comparable limitations."

Claims 4 and 20

Appellant's dependent claims 4 and 20 are distinguished from Muttik for the reasons given above with respect to their base claims 1 and 16. As should be evident from the discussion above with respect to the differences between Edwards and the limitations of the base claims 1 and 16, the limitations of the base claims 1 and 16 are not found in Edwards either. Nor would a proper combination of Muttik and Edward result in the subject matter of appellant's base claims

1 and 16. Appellant's base claim 1, for example, calls for combining on-demand anti-virus scan requests and on-access anti-virus scan requests in a virus scan request queue, and distributing the on-demand anti-virus scan requests and the on-access anti-virus scan requests from the virus scan request queue to the virus checkers (for on-demand anti-virus scanning concurrent with on-access anti-virus scanning).

Muttik is dealing with a problem of the storage of ever increasing amounts of data leading to scans taking longer and longer. Muttik addresses this problem by scanning the files in a circular manner such that when all files have been scanned the scanner automatically starts the process again at the first file. Any new files created during a scan while therefore take their place with other files in the list of files to be scanned and, given the circular nature of the scan, will themselves in time be scanned. (Muttik, paragraph [0010].) New viruses discovered during a scan are checked against subsequently scanned files and against early files when scanning automatically starts from the beginning again. (Muttik, paragraph [0011].)

Edwards teaches that a particular problem with the Edwards FIG. 1 prior art on-access anti-virus scanner can be solved by adding a priority queue 220 to produce the on-access antivirus scanner shown in Edwards FIG. 2.

"[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." In re Kahn, 441 F. 3d 977, 988 (Fed. Cir. 2006). A fact finder should be aware of the distortion caused by hindsight bias and must be cautious of arguments reliant upon ex post reasoning. See KSR v. Teleflex, 550 U.S. __ (2007), citing Graham,

383 U. S. at 36 (warning against a "temptation to read into the prior art the teachings of the invention in issue" and instructing courts to "guard against slipping into the use of hindsight.").

The problem that the inventor is trying to solve must be considered in determining whether or not the invention would have been obvious. The invention as a whole embraces the structure, properties and problems it solves. In re Wright, 848 F.2d 1216, 1219, 6 U.S.P.Q.2d 1959, 1961 (Fed. Cir. 1988). In contrast to Muttik and Edwards, the appellant's specification teaches that it is desirable to use the same virus checkers for on-demand and on-access virus checking. This leads to several problems because the priority and workload for the on-demand scan requests are so much different from the priority and workload for the on-access scan requests. The different priority and workload would suggest that the on-demand scan requests should be treated differently from the on-access scan requests. Nevertheless, the appellant's specification teaches that it is desirable to combine the on-demand requests and the on-access requests in a queue under appropriate conditions. This is explained in appellant's specification on page 11 line 20 to page 12 line 23 as follows:

> **[00030]** The scanning task shown in FIG. 2 is generally referred to as an "on-access" virus scan. An "on-access" virus scan is processed in real-time when scanning is triggered by user-initiated file access. Another kind of virus scan is known as an "on-demand" virus scan. An "on-demand" virus scan is scheduled at a lower priority than "on-access," and it typically involves scanning all files of virus-checkable file type in a one or more specified file systems. For example, "on-demand" virus checking is scheduled when a new virus is discovered, when new unchecked files are migrated into a file server, or prior to archiving or

backing-up unchecked files. Although lower in priority from a scheduling point of view, "on-demand" virus checking has often been more burdensome on the data processing system than "on-access" virus checking. A full file system scan may generate a much more intense scanning load when a multitude of files in the file system must be scanned. Even though this scanning workload is distributed over multiple virus checkers, the volume of scans will generate a significant resource load on the operating system of the data mover. Moreover, it is desirable to fully utilize the capabilities of the virus checkers in order to complete the full file system scan as soon as possible

[00031] In order to mitigate any general performance degradation on the data mover during user file access, it is desirable to mix "on-demand" virus scan requests with "on-access" virus scan requests in a shared virus scan request queue. For example, outstanding "on-demand" virus scan requests are added to the shared queue when the number of requests in the shared queue falls below a threshold. The threshold is selected to provide a relatively continuous flow of requests to the virus checkers without significantly degrading the response time of the virus checkers for responding to the "on-access" requests. Moreover, it is desirable to add outstanding "on-demand" virus scan requests to the shared queue in manageable "chunks", and to wait until the virus scan requests in each chunk have been serviced before sending another chunk of "on-demand" virus scan requests.

Neither Muttik nor Edwards suggest these problems or the particular solution as defined in appellant's claim 1. The fact that the on-demand scan requests have priority and workloads that are so much different from the priority and workloads of the on-access scan requests, teaches away from treating the on-demand scan requests and the on-access scan requests in a similar fashion by combining them in a virus scan request queue for distribution to virus checkers.

Neither Muttik nor Edwards says anything to the contrary. Edwards in fact refers to an on-access virus scanner as being a type of virus scanner different from an on-demand virus scanner (see col. 1, lines 43-47) and having particular problems (col. 1 line 65 to col. 2 line 67) and scanning data in another context than an on-demand virus scanner (col. 10 lines 3-7).

In short, the teachings of Muttik and Edwards and the nature of the problem solved by the appellant show that the subject matter of appellant's base claims 1 and 16 would not have been obvious from Muttik and Edwards. Therefore, the subject matter of appellant's dependent claims 4 and 20 would not have been obvious from Muttik and Edwards.

Claims 5 and 21

Appellant's dependent claims 5 and 21 are distinguished from Muttik as discussed above with respect to their respective base claims 1 and 16, and they are patentable over the proposed combination of Muttik and Edwards as discussed above with respect to claims 4 and 20.

Appellant's dependent claims 5 and 21 further define that the on-access anti-virus scan requests are given priority over the on-demand anti-virus scan requests by inhibiting the placement of on-demand anti-virus scan requests onto the virus scan request queue when the number of anti-virus scan requests on the virus scan request queue reaches a threshold, and not inhibiting the placement of on-access anti-virus scan requests onto the virus scan request queue when the number of anti-virus scan requests on the virus scan request queue reaches the threshold.

Page 8 of the Official Action cites Muttik for teaching that on-access anti-virus scan requests are given priority over the on-demand anti-virus scan requests. However, as discussed above with respect to claim 1, appellant's view is that Muttik paragraph [0037] does not teach placing on-demand anti-virus scan requests on the queue of on-demand virus scan requests.

Page 8 of the Official Action cites Edwards (Col. 5, lines 66-67, col. 6, lines 1-3) for teaching that certain scan requests are given priority based on certain characteristics by not inhibiting a first kind of scan request while inhibiting the placement of a second kind of scan request based on a second group of characteristics with the number of anti-virus scan request on the virus scan request queue reaches a threshold. In particular, Edwards teaches "a pending scan request from user A may be determined to be more suitable than a pending scan request from user B if three of the four scanner threads are already scanning scan requests from user B. This prevents a single user B from monopolizing the virus scanner (Column 5, lines 66-67, Column 6, lines 1-3)." Other passages of Edwards disclose that event filter 110 or scan prioritizer 240 places the on-access virus scan requests on the virus scan request queue, and some of the virus scan requests on the queue are given priority over other virus scan requests on the priority queue because the event filter or scan prioritizer places the virus scan requests on the queue in the order that the threads should process them, and because the scanner threads select for scanning particular virus scan requests that are on the queue based on the position of the virus scan requests on the queue or the operational characteristics of the virus scan requests. For example, Edwards col. 5, lines 39-43 say: "In one embodiment, the characteristics obtained by the pre-processor threads 210 are used by the event filter 110 to prioritize the scan requests by placing

them in the priority queue 220 in the order in which the scanner threads 120 should process them." Edwards col. 6 lines 23-29 say: "As another example, using the scan request's operational characteristics, such as a time stamp of when the scan request was triggered by the event filter 110 or when it was placed on the priority queue 220, scan requests that have been passed over too often (i.e. that have been on the priority queue 220 the longest) could eventually be given higher priority than scan requests that would otherwise come first." Edwards col. 6 lines 42-46 say: "In each of the above examples, the scanner threads 120 process scan requests either by selecting the most suitable pending request, or by selecting the next pending request that was placed on the priority queue in the optimal priority order by the scan prioritizer 240."

However, neither Muttik nor Edwards discuss combining on-demand anti-virus scan requests and on-access anti-virus scan request in a virus scan request queue so that on-access anti-virus scan requests that are put in the queue would be given priority over on-demand anti-virus scan requests that are put in the queue. Nor is there any disclosure in Edwards col. 5, lines 66-67 or col. 6, lines 1-3 of inhibiting the placement of any particular kind of virus scan request onto the virus scan request queue when the number of anti-virus scan requests on the queue reaches a threshold. Inhibiting placement of a request onto the virus scan request queue is different from placing a request on the virus scan request queue in a particular order with respect to requests already on the virus request queue. Inhibiting placement of a request onto the virus scan request queue is also different from a thread selecting a particular request on the virus scan queue for distribution to one of the virus checkers.

Claims 12, 14, 24, and 26

Appellant's independent claim 12 includes limitations found in appellant's claims 1, 4, 6, and 7. In a similar fashion, appellant's claim 24 includes limitations found in appellant's claims 16, 20, 22, and 23, and also recites "consecutively" placing the chunks of on-demand anti-virus scan requests onto the virus scan request queue.. Therefore, appellant's independent claim 12 is patentable over the proposed combination of Muttik and Edwards for the reasons given above with reference to claims 4, 6, and 7, and appellant's independent claim 24 is patentable over the proposed combination of Muttik and Edwards for the reasons given above with reference to claims 20, 22, and 23.

Claims 13 and 25

Appellant's claims 13 and 25 are dependent on claims 12 and 24, and therefore are patentable over the proposed combination of Muttik and Edwards for the reasons given above with reference to claims 12 and 24. In addition, claims 13 and 25 explicitly define that the on-access anti-virus scan requests are produced in response to user access to files. As discussed above with respect to claim 1, Muttik fails to disclose that the adding of the new files to the system mid-scan or the placing of these new files at a high position in the queue is done in response to user access to the new files.

Claim 27

Appellant's dependent claim 27 is dependent upon claim 24 and therefore is patentable over the proposed combination of Muttik and Edwards for the reasons given above with reference to claim 25. Claim 27 further recites limitations similar to those found in appellant's claim 21, and therefore is patentable over the proposed combination of Muttik and Edwards for the reasons given above with reference to claim 21.

Claims 15 and 28

Appellant's claims 15 and 28 are dependent upon appellant's claims 12 and 24, respectively, and therefore are patentable over the proposed combination of Muttik and Edwards for the reasons given above with reference to claims 12 and 24. In addition, claims 15 and 28 further recite "inhibiting the placement of at least one of the chunks onto the virus scan request queue until completion of anti-virus scanning of the anti-virus scan requests in a prior one of the chunks." Page 13 of the Official Action cites paragraph [0037] of Muttik as "inhibiting the on-demand scan's placement in the queue by first scanning files as a result of an access to a file." However, claims 15 and 28 recite inhibiting the placement of at least one of the chunks onto the virus scan request queue, and not inhibiting placement in the queue. Moreover, Muttik paragraph [0037] is placing a new scan request at a high position in the queue, and not inhibiting placement of other requests onto the queue. Placement of a new request on the queue before an old request already on the queue is different from inhibiting the placement of the old request onto

the queue. (Moreover, as discussed above with respect to claim 4, Muttik paragraph [0037] does not disclose that the new request is a result of an access to a file.)

Claim 17

Claim 17 is dependent upon claim 16, and therefore is patentable over the proposed combination of Muttik and Edwards for the reasons discussed above with reference to claims 16 and 20.

In view of the above, the rejection of claims 1-28 should be reversed.

Respectfully submitted,

/Richard C. Auchterlonie/

Richard C. Auchterlonie
Reg. No. 30,607
NOVAK DRUCE & QUIGG, LLP
1000 Louisiana, 53rd Floor
Houston, TX 77002
713-571-3400 (telephone)
713-456-2836 (telefax)

VIII. CLAIMS APPENDIX

The claims involved in this appeal are as follows:

1. A method of operating a plurality of virus checkers for on-demand anti-virus scanning concurrent with on-access anti-virus scanning, the method comprising:

combining on-demand anti-virus scan requests and on-access anti-virus scan requests in a virus scan request queue; and

distributing the on-demand anti-virus scan requests and the on-access anti-virus scan requests from the virus scan request queue to the virus checkers.

2. The method as claimed in claim 1, wherein the on-access anti-virus scan requests are produced in response to user access to files.

3. The method as claimed in claim 1, wherein the on-demand anti-virus scan requests are produced in response to a system administrator requesting a scan of files within a specified file system.

4. The method as claimed in claim 1, wherein a pool of threads distribute the on-demand anti-virus scan requests and the on-access anti-virus scan requests from the virus scan request

queue to the virus checkers, each anti-virus scan request on the virus scan request queue being serviced by a respective one of the threads in the pool of threads.

5. The method as claimed in claim 1, wherein the on-access anti-virus scan requests are given priority over the on-demand anti-virus scan requests by inhibiting the placement of on-demand anti-virus scan requests onto the virus scan request queue when the number of anti-virus scan requests on the virus scan request queue reaches a threshold, and not inhibiting the placement of on-access anti-virus scan requests onto the virus scan request queue when the number of anti-virus scan requests on the virus scan request queue reaches the threshold.

6. The method as claimed in claim 1, which includes grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and placing the chunks onto the virus scan request queue.

7. The method as claimed in claim 6, which includes inhibiting the placement of at least one of the chunks onto the virus scan request queue until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks.

8. A method of operating a plurality of virus checkers, the method comprising:

distributing on-demand anti-virus scan requests and on-access anti-virus scan requests to the virus checkers so that the virus checkers perform on-demand anti-virus scanning concurrent with on-access anti-virus scanning;

which includes grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and for each chunk, distributing the multiple ones of the on-demand anti-virus scan requests over the virus checkers.

9. The method as claimed in claim 8, wherein the on-access anti-virus scan requests are produced in response to user access to files.

10. The method as claimed in claim 8, wherein the on-demand anti-virus scan requests are produced in response to a system administrator requesting a scan of files within a specified file system.

11. The method as claimed in claim 8, which includes inhibiting the distribution of the multiple ones of the on-demand anti-virus scan requests from at least one of the chunks to the virus checkers until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks.

12. A method of operating a plurality of virus checkers for on-demand anti-virus scanning concurrent with on-access anti-virus scanning, the method comprising:

combining on-demand anti-virus scan requests and on-access anti-virus scan requests in a virus scan request queue; and

a pool of threads distributing the on-demand anti-virus scan requests and the on-access anti-virus scan requests from the virus scan request queue to the virus checkers, each anti-virus scan request on the virus scan request queue being serviced by a respective one of the threads in the pool of threads,

which includes grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and for each chunk, checking whether the number of anti-virus scan requests on the virus checking queue is less than a threshold, and upon finding that the number of anti-virus scan requests on the virus checking queue is less than the threshold, placing said each chunk on the virus scan request queue.

13. The method as claimed in claim 12, wherein the on-access anti-virus scan requests are produced in response to user access to files.

14. The method as claimed in claim 12, wherein the on-demand anti-virus scan requests are produced in response to a system administrator requesting a scan of files within a specified file system.

15. The method as claimed in claim 12, which includes inhibiting the placement of at least one of the chunks onto the virus scan request queue until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks.

16. A virus checking system comprising:

a plurality of virus checkers for on-demand anti-virus scanning concurrent with on-access anti-virus scanning;

a virus scan request queue; and

at least one processor coupled to the virus checkers and the virus scan request queue for sending virus scan requests from the virus scan request queue to the virus checkers, said at least one processor being programmed for placing on-demand anti-virus scan requests and on-access anti-virus scan requests onto the virus scan request queue, and for distributing the on-demand anti-virus scan requests and the on-access virus scan requests from the virus scan request queue to the virus checkers.

17. The virus checking system as claimed in claim 16, wherein said at least one processor and said virus scan request queue are in a file server, and the virus checkers are separate from the file server.

18. The virus checking system as claimed in claim 16, wherein said at least one processor is programmed to place each on-access request onto the virus scan request queue in response to user access of a respective file.

19. The virus checking system as claimed in claim 16, wherein said at least one processor is programmed to produce the on-demand anti-virus scan requests in response to a system administrator requesting a scan of files within a specified file system.

20. The virus checking system as claimed in claim 16, wherein said at least one processor is programmed to execute multiple threads for distributing the on-demand anti-virus scan requests and the on-access anti-virus scan requests from the virus scan request queue to the virus checkers, each anti-virus scan request on the virus scan request queue being serviced by a respective one of the threads in the pool of threads.

21. The virus checking system as claimed in claim 16, wherein said at least one processor is programmed for giving the on-access anti-virus scan requests priority over the on-demand anti-virus scan requests by inhibiting the placement of on-demand anti-virus scan requests onto the virus scan request queue when the number of anti-virus scan requests on the virus scan request queue reaches a threshold, and not inhibiting the placement of on-access anti-virus scan requests onto the virus scan request queue when the number of anti-virus scan requests on the virus scan request queue reaches the threshold.

22. The virus checking system as claimed in claim 16, wherein said at least one of the processors is programmed for grouping the on-demand anti-virus scan requests into chunks, each of the chunks including multiple ones of the on-demand anti-virus scan requests, and placing the chunks onto the virus scan request queue.

23. The virus checking system as claimed in claim 22, which includes inhibiting the placement of at least one of the chunks onto the virus scan request queue until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks.

24. A virus checking system comprising:

a plurality of virus checkers for on-demand anti-virus scanning concurrent with on-access anti-virus scanning; and

a file server coupled to the virus checkers for sending virus scan requests to the virus checkers, the file server including a virus scan request queue, and the file server being programmed for placing on-demand anti-virus scan requests and on-access anti-virus scan requests onto the virus scan request queue; and for executing multiple threads for distributing the on-demand anti-virus scan requests and the on-access anti-virus scan requests from the virus scan request queue to the virus checkers, each anti-virus scan request on the virus scan request queue being serviced by a respective one of the threads in the pool of threads, the file server further being programmed for grouping the on-demand anti-virus scan requests into chunks, each

of the chunks including multiple ones of the on-demand anti-virus scan requests, and for consecutively placing the chunks onto the virus scan request queue.

25. The virus checking system as claimed in claim 24, wherein the file server is programmed for producing the on-access anti-virus scan requests in response to user access to files.

26. The virus checking system as claimed in claim 24, wherein the file server is programmed to produce the on-demand anti-virus scan requests in response to a system administrator requesting a scan of files within a specified file system.

27. The virus checking system as claimed in claim 24, wherein the file server is programmed for checking for each chunk whether the number of anti-virus scan requests on the virus checking queue is less than a threshold, and upon finding that the number of anti-virus scan requests on the virus checking queue is less than the threshold, placing said each chunk on the virus scan request queue.

28. The virus checking system as claimed in claim 24, wherein the file server is programmed for inhibiting the placement of at least one of the chunks onto the virus scan request queue until completion of anti-virus scanning for the anti-virus scan requests in a prior one of the chunks.

IX. EVIDENCE APPENDIX

None.

X. RELATED PROCEEDINGS APPENDIX

None.